

**This Page Is Inserted by IFW Operations  
and is not a part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-3306

(43)公開日 平成11年(1999)1月6日

(51)Int.Cl.<sup>6</sup>  
G 0 6 F 13/00  
12/00  
15/16

識別記号  
3 5 5  
5 3 1  
3 7 0

F I  
G 0 6 F 13/00  
12/00  
15/16

3 5 5  
5 3 1 J  
3 7 0 Z

審査請求 有 請求項の数12 OL (全 14 頁)

(21)出願番号 特願平9-154811

(22)出願日 平成9年(1997)6月12日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 坂倉 隆史

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

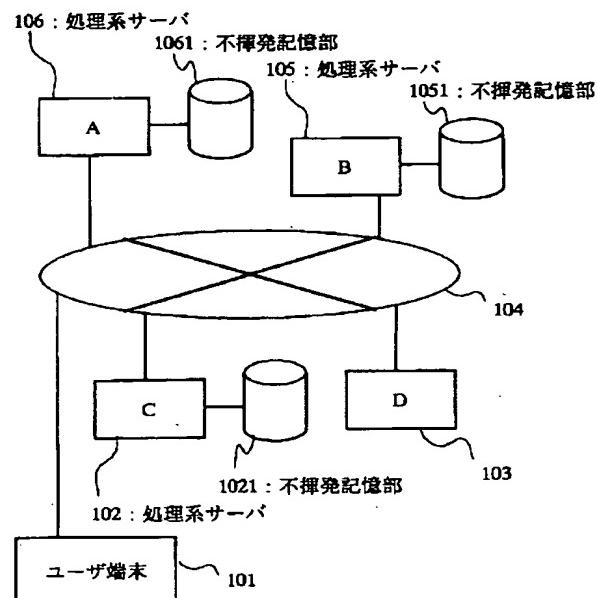
(74)代理人 弁理士 宮田 金雄 (外2名)

(54)【発明の名称】 エージェント方式

(57)【要約】

【課題】 プログラムとデータをともに送信し、送信先で実行するエージェント方式において、ミッションクリティカルなアプリケーションの実行を可能とするエージェントの再実行、取り消し手段を提供する。

【解決手段】 エージェントを処理する処理系に、エージェント自身やエージェントを実行したオペレーションログを格納するための不揮発記憶領域を設ける。



**【特許請求の範囲】**

**【請求項1】** 以下の要素を有するエージェント方式  
 (a) データと上記データを処理対象として処理する手続きとを有するエージェントを生成して送信するユーザ端末、(b) 上記ユーザ端末が送信する上記エージェントを受信して受信したエージェントを実行し上記手続きに従って上記データを処理する処理系と、  
 上記エージェントが利用できる不揮発記憶領域とを備えたサーバ。

**【請求項2】** 上記処理系は、上記エージェントの実行状態を示すオペレーションログを採取することを特徴とする請求項1に記載のエージェント方式。

**【請求項3】** 上記処理系は、受信したエージェントと上記オペレーションログとの少なくともいずれかを上記不揮発記憶領域に記憶することを特徴とする請求項2に記載のエージェント方式。

**【請求項4】** 上記処理系は、上記オペレーションログを上記ユーザ端末に返送することを特徴とする請求項2に記載のエージェント方式。

**【請求項5】** 上記処理系は、受信したエージェントと上記オペレーションログの両方が上記不揮発記憶領域に記憶されている場合に、上記不揮発記憶領域に記憶された上記オペレーションログを参照して、上記不揮発記憶領域に記憶されたエージェントを用いて、エージェントの再実行を行なうことを特徴とする請求項3に記載のエージェント方式。

**【請求項6】** 上記処理系は、上記オペレーションログが上記不揮発記憶領域に記憶されている場合に、上記不揮発記憶領域に記憶された上記オペレーションログを参照して上記ユーザ端末にエージェントの再発行を要求することにより、エージェントの再実行を行なうことを特徴とする請求項3に記載のエージェント方式。

**【請求項7】** 上記ユーザ端末は、上記返送されたオペレーションログを参照して、上記オペレーションログに対応するエージェントを再発行することにより、エージェントの再実行を行なうことを特徴とする請求項4に記載のエージェント方式。

**【請求項8】** 上記ユーザ端末は、上記オペレーションログを参照して、上記エージェントを削除することを特徴とする請求項3に記載のエージェント方式。

**【請求項9】** 上記ユーザ端末は、先に発行したエージェントの追跡を行なう追跡用エージェントを発行し、上記追跡用エージェントは上記オペレーションログを参照してエージェントの追跡を行なうことを特徴とする請求項2に記載のエージェント方式。

**【請求項10】** 上記追跡用エージェントは、先に発行されたエージェントの追跡を行なうとともに、先に発行されたエージェントに対応して記録されたオペレーションログの消去を行なうことを特徴とする請求項9に記載のエージェント方式。

**【請求項11】** ネットワークを介して接続可能な複数のサーバ上の複数の処理系を送信先としてエージェントを送信可能なユーザ端末と、上記複数の処理系の属性に関する情報を属性情報として保持する属性情報サーバとを備えたエージェント方式であって、上記エージェントは、上記属性情報サーバが保持する属性情報を参照して実行する処理系を選択することを特徴とするエージェント方式。

**【請求項12】** データと上記データを処理対象として処理する手続きとを有するエージェントを生成して送信するユーザ端末と、

上記ユーザ端末が送信する上記エージェントを受信して、上記エージェントを実行し、上記手続きに従って上記データを処理するとともに、上記エージェントを実行したオペレーションログを上記ユーザ端末に返送する処理系を備えたサーバとからなるエージェント方式。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】** この発明は、多種多様の計算機が相互に接続された多様な通信環境において、手順、データを含むオブジェクトをエージェントとしてある計算機に送信し、オブジェクトを送信先の計算機で実行するエージェント方式に関する。

**【0002】**

**【従来の技術】** 近年“エージェント”と呼ばれるソフトウェア技術が知られてきている。エージェント方式とは、プログラム（処理の手順、手続きともいう）とデータを持ったオブジェクト（エージェント、ネットワークエージェントともいう）をある計算機に送信し、実行するという処理形態をとるものである。本明細書において、エージェントとは、この処理方式、又は、オブジェクトを指すものとする。本明細書で述べるエージェントは、いわゆる一般的なエージェントでも、計算機内で常駐し、ある一定の目的を遂行する広義のソフトウェアエージェントでもない。

**【0003】** 日本特許出願の特開平7-182174号（米国特許出願S/N 08/090521、出願日1993年7月8日の対応出願）「リモートプログラミングの実施方法」には、米国のジェネラルマジック社（General Magic, Inc.）が開発したテレスクリプト（Telescript、ジェネラルマジック社の商標）と呼ばれるエージェント記述用言語の仕様が詳細に記載されている。

**【0004】** 日本特許出願の公開公報特開平7-182174号を引用して、従来の技術を説明する。図16は、従来技術によるコンピュータシステムの構造を示す図である。図17は、リモートプログラミングを使用する従来技術による方法のフローチャートである。図18は、リモートプログラミングを具体化する従来技術によるネットワークを示す図である。

【0005】日本国特許特開平7-182174号(12ページ、カラム21、27~32行目)によれば、マスメモリ917A、例えば、磁気ディスクや磁気テープは、プログラム、データ又はCPU910によって直ちには必要とされないか若しくはメインメモリ917Bの大きさの制限のためにメインメモリ917B中に収容することのできないプログラム又はデータの一部分を記憶するために用いられる。

【0006】また、日本国特許特開平7-182174号(15ページ、カラム27、20~23行目)では、Wolfsen等によって開示されたシステムを開示している。そのシステムは、「あるネットワーク内においてあるプロセスが1つのコンピュータシステムから別のコンピュータシステムに移行することのできるシステム(15ページ、カラム27、7~9行目)」であり、そのシステムにおいては、「～プロセスは、それぞれのプロセスが実行されているコンピュータシステムのマスメモリにデータを直接に記憶させたりこのマスメモリからデータを直接検索したりする(15ページ、カラム27、20~3行目)。」

【0007】また、図17に示すフローチャートは、リモートプログラミングの具体例を示している。(中略)具体例をリモートプログラミング環境において実現するには、クライアントプロセス9352(図18)は、ステップ931(図17)においては、このステップ931の詳細なフローチャートに示すインストラクションから構成されているプログラムを作成する。このステップ931の詳細については、以下に説明する。処理は、ステップ931からステップ932に進み、ステップ932において、プログラムは、矢印9358で示すように、ネットワーク9356を介しコンピュータ930B(図18)に転送される。処理は、ステップ932(図17)からステップ933に進み、このステップ933において、プログラムは、コンピュータ930Bによって実行される。この実行の中には、プログラムは、コンピュータ930B上のプロセス9352A(図18)となっている。そして、プロセス9352Aにおいて、プログラムのインストラクションは、ステップ931のフローチャートに従って実行される。ステップ931-B(図17)において、ファイル名のリストが作成される。(中略)ステップ931-Gにおいて、サーバプロセス9354(図18)に適切なインストラクションを送出することによってファイルが削除される。処理は、ステップ931-G(図17)からステップ931-Jを介して、ステップ931-Cに戻る。リスト中の全てのファイル名が処理されたならば、処理は、ステップ931-Cからステップ931-Kに進み、ここでプロセス9352Aが完了する。矢印9360(図18)に示すように、プロセス9352Aとサーバプロセス9354間の全ての相互作用は、ネットワーク9356を使用

することなく、全てコンピュータ930B中において起る。プログラムが成功のうちに終了した後、処理は、ステップ933からステップ934(図17)に移行し、このステップ934において、サーバプロセス9354(図18)は、矢印9362によって示すようにプログラムが成功のうちに終了したことをクライアントプロセス9352に報告する。このリモートプログラミングプロセッサーがネットワーク通信メディアをただ2回しか使用していないことに注意されたい。第1回の使用は、矢印9358によって示すように、インストラクションリスト又はプログラムをサーバプロセス9354に送出する使用であり、第2の使用は、矢印9362によって示すように、プログラムが成功のうちに終了したことの通知をサーバプロセス9354から受ける使用である(14ページ、カラム25~26)。

【0008】このように、従来の技術で、「プロセスは、それぞれのプロセスが実行されているコンピュータシステムのマスメモリにデータを直接に記憶させたり、このマスメモリからデータを直接検索したりする」点が述べられているが、「ここでのマスメモリへのデータの記憶は、プロセスの処理対象となるデータの記憶を指している」とするのが妥当であり、プロセス自身やプロセスの実行に関連する管理情報を記憶することについては述べられていない。また、リモートプログラミングプロセッサーにおいては、サーバプロセス9354は、プログラムが成功のうちに終了したことをクライアントプロセス9352に報告する。この報告は、プログラムの終了状態が正常か否かを報告することと解するのが妥当であり、プログラムの実行状態をトレースし、逐一クライアントに報告することは述べられていない。また、トレースした実行状態を記憶させることも述べられていない。また、リモートプログラムの送出やリモートでのプログラム実行については述べられているが、リモートプログラムをサーバの補助記憶装置(マスメモリ)に記憶することは述べられていない。

【0009】このように、従来のエージェント方式においては、ディスクにオブジェクト(エージェント)を格納していかなかったので、処理系はエージェントの実行を正常終了しなかった場合に、そのエージェントを再実行することはできなかった。また、従来のエージェント方式においては、ディスクにエージェントを実行した結果であるログデータを記憶していかなかったので、ログデータを参照して、実行結果を問い合わせることはできなかった。また、エージェントが送信された経路を調べることもできなかった。また、実行中に中断されたエージェントの再実行を要求することもできなかった。また、ログデータを調べてデータベースのリカバリを行う再試行メソッドも持っていないかった。そのため、ミッションクリティカルな業務にエージェントを用いることはできなかった。

## 【0010】

【発明が解決しようとする課題】従来の汎用機を中心におくセントラライズドシステムやクライアント／サーバ型システムとは異なり、エージェントはプログラムとデータを持ったオブジェクトをある計算機に送信し、実行するという処理形態をとる。そのため、エージェント技術によると、ユーザオペレーションは、同期的には処理されない。同期的に処理されないため、もし、プログラム実行中に障害が発生しても、ユーザは、障害の発生やその障害の内容を知ることができない。エージェントをインターネットに接続されたサーバから参照用データを集めるとといった用途に用いる場合では、仮に、エージェントが消失しても、被害はユーザにとってそれほど深刻ではないが、データベースにトランザクション処理を行うようなエージェントプログラムで障害が発生すると、ユーザにとっての被害は甚大である。逆に、このような欠点があるため、エージェント技術は重要なトランザクション処理を含むようなプログラムやサービスには、利用されてこなかった。

【0011】この発明は、上記のような問題点を解決するためになされたものであり、エージェントが信頼性を必要とするミッションクリティカルな処理を行うことが可能なエージェント方式を実現することを目的としている。また、エージェントの実行状態を示すオペレーションログを採取し、採取したオペレーションログを用いて、中断されたエージェントの再実行や発行されたエージェントの取り消しが可能なエージェント方式を実現することを目的としている。

## 【0012】

【課題を解決するための手段】この発明に係るエージェント方式は、以下の要素を有することを特徴とする。

(a) データと上記データを処理対象として処理する手続きとを有するエージェントを生成して送信するユーザ端末、(b) 上記ユーザ端末が送信する上記エージェントを受信して受信したエージェントを実行し上記手続きに従って上記データを処理する処理系と、上記エージェントが利用できる不揮発記憶領域とを備えたサーバ。

【0013】上記処理系は、上記エージェントの実行状態を示すオペレーションログを採取することを特徴とする。

【0014】上記処理系は、受信したエージェントと上記オペレーションログとの少なくともいずれかを上記不揮発記憶領域に記憶することを特徴とする。

【0015】上記処理系は、上記オペレーションログを上記ユーザ端末に返送することを特徴とする。

【0016】上記処理系は、受信したエージェントと上記オペレーションログの両方が上記不揮発記憶領域に記憶されている場合に、上記不揮発記憶領域に記憶された上記オペレーションログを参照して、上記不揮発記憶領域に記憶されたエージェントを用いて、エージェントの

再実行を行なうことを特徴とする。

【0017】上記処理系は、上記オペレーションログが上記不揮発記憶領域に記憶されている場合に、上記不揮発記憶領域に記憶された上記オペレーションログを参照して上記ユーザ端末にエージェントの再発行を要求することにより、エージェントの再実行を行なうことを特徴とする。

【0018】上記ユーザ端末は、上記返送されたオペレーションログを参照して、上記オペレーションログに対応するエージェントを再発行することにより、エージェントの再実行を行なうことを特徴とする。

【0019】上記ユーザ端末は、上記オペレーションログを参照して、上記エージェントを削除することを特徴とする。

【0020】上記ユーザ端末は、先に発行したエージェントの追跡を行なう追跡用エージェントを発行し、上記追跡用エージェントは上記オペレーションログを参照してエージェントの追跡を行なうことを特徴とする。

【0021】上記追跡用エージェントは、先に発行されたエージェントの追跡を行うとともに、先に発行されたエージェントに対応して記録されたオペレーションログの消去を行なうことを特徴とする。

【0022】この発明に係るエージェント方式は、ネットワークを介して接続可能な複数のサーバ上の複数の処理系を送信先としてエージェントを送信可能なユーザ端末と、上記複数の処理系の属性に関する情報を属性情報として保持する属性情報サーバとを備えたエージェント方式であって、上記エージェントは、上記属性情報サーバが保持する属性情報を参照して実行する処理系を選択することを特徴とする。

【0023】この発明に係るエージェント方式は、データと上記データを処理対象として処理する手続きとを有するエージェントを生成して送信するユーザ端末と、上記ユーザ端末が送信する上記エージェントを受信して、上記エージェントを実行し、上記手続きに従って上記データを処理するとともに、上記エージェントを実行したオペレーションログを上記ユーザ端末に返送する処理系を備えたサーバとからなることを特徴とする。

## 【0024】

## 【発明の実施の形態】

実施の形態1．本発明を適用した好適な実施の形態につき説明する。この実施の形態では、データベースを更新するトランザクション処理を行うエージェントの場合を説明するが、トランザクション処理を含まないエージェントに、この発明を適用しても構わない。本発明は、エージェントが処理系上で実行され、処理系はあるオペレーティングシステム上で、又はオペレーティングシステムとして実行されるシステムを想定している。処理系は、システムが提供すべき機能をエージェントに提供するが、その1つとして不揮発記憶領域を用意する。この

発明のエージェント方式の具体的な実現方法としては、エージェントをオブジェクトとしてカプセル化し、計算機から計算機へ、処理系から処理系への移動をサポートするメカニズムを想定している。実際の実現方法は、どんな方法でも構わない。また、この発明を実施するに際しては、エージェントの移動を実現する具体的な実現方法はどんな方法でも構わない。この発明は、エージェントの送受信を行う送受信手段の実現方法に依存するものではない。エージェントの移動に用いられる通信プロトコルの種類も問わない。また、この明細書では、1つの計算機上に処理系が1つだけ存在する場合には、エージェントが1つの処理系から他の処理系へ移動するという表現と、エージェントが1つの計算機から他の計算機へ移動するという表現は、実質的には同じ動作を指すものとする。また、所定の通信プロトコルで用いる書式の特定フィールドに設定される値が、処理系をデスティネーションとする移動と、計算機をデスティネーションとする移動とで異なっても構わない。いずれもこの発明の実施には影響を与えないものとする。

【0025】図1は、本実施の形態のエージェント方式が稼働するシステム構成を示す図である。エージェントは、サーバ上の処理系で実行される。104はエージェントが実行される処理系を持つサーバが接続されるネットワークである。例えば、インターネットで用いられるようなIPプロトコルが通過するネットワークである。101はエージェントを生成し送出するユーザ端末、102、103、105、106はそれぞれ処理系が動作するサーバシステム（処理系サーバ）である。エージェントは、ネットワーク104上をTCP/IP（Transmission Control Protocol / Internet Protocol）により移動する。この実施の形態では、TCP/IPのプロトコルの一種であるUDP（User Datagram Protocol）を用いるものとする。1021、1051、1061は処理系サーバに接続され、処理系で実行されるエージェントから利用可能な不揮発記憶部である。これらの不揮発記憶部は、到着したエージェントを格納可能である。また、エージェントを実行したときの実行状態を記憶可能である。ユーザ端末101において、エージェントを送出するプログラムは、例えば、JAVA言語で作成されたJAVAアプレットであり、WWW（World Wide Web）サイトよりダウンロードされる。

【0026】ここで、システム動作の説明に先だって、この発明が適用された処理系の不揮発記憶メカニズムについて説明する。図2に、構成を示す。201は処理系202が動作するサーバシステム、203は処理系がエージェントへの不揮発記憶領域提供のために利用する不揮発記憶部（ディスク）である。本実施の形態において、当不揮発記憶領域提供メカニズムは、ごく単純な領

域管理を行っている。処理系が到着したエージェントをディスクに格納する場合、また、ユーザ、つまり、エージェントからディスクへのデータの書き込み要求があると、データ（エージェントの場合もある）は、単純にディスク領域の後方に追加され、書き込みポインタが更新される。最後方にある一定のサイズ以上の空き領域があると、不要領域の先頭にポインタを戻して、不要領域の解放をする。ところで、処理系は、通常、ディスク書き込みの際、システムの用意するディスク書き込み機能を使用するが、この発明の不揮発記憶部は、RAWディスクインターフェースを通してアクセスする。RAWディスクインターフェースとは、上位のファイルシステムを介さずに、例えば、ドライバレベルでディスクに直接読み書きを行うインターフェースである。上位のファイルシステムを介在させると、ファイルに対する書き込みは完了していても、ディスクへの書き込みが完了していない場合が発生することがあり、そのときにサーバシステムがダウンしたりすると、ファイルに書いたデータとディスクにかかれたはずのデータの不整合が発生して、データの内容が保証されなくなる。そのため、この実施の形態では、エージェントが利用するディスクに、RAWディスクインターフェースを通してアクセスする。これにより、エージェントからこの不揮発記憶部への書き込み要求は同期的に処理される。書き込みの正常終了報告が処理系からあれば、確実に不揮発記憶部中に記録され、正常終了報告がない場合、書き込み処理中にシステムに障害が発生したときは、書き込み前のデータ内容が保証される。

【0027】図3は、ネットワークエージェントシステムにおいて使用されるエージェントの構成の一例を示す図である。図3において、141がエージェントである。エージェント141は、エージェントを識別するエージェントのID1411とエージェントの種類1412と端末のユーザを識別するユーザID1415とユーザID1416に対するパスワード1416とエージェント141がサーバシステムにおいて行う処理の手続き（手順ともいう）1419とサーバシステムにおける処理の結果エージェントに添付されるプログラム、データ1420を含んでいる。手続き1419は、各サーバシステムに備えられた手続き解読手段のいずれもが解読できる中間言語形式で記述されている。エージェント141がデータベースを検索するための検索エージェントの場合は、手続き1419に検索の対象となるデータベースの名前や検索の条件などのデータベースを検索する手続きが記述される。また、エージェント141が電子メールを受信するための電子メール受信エージェントの場合は、手続き1419にメールを受信したいメールサーバの名前とメールを受信する端末で表示可能なメールの種類の情報が記述される。

【0028】この発明のエージェントは、手続き中にエ

ージェントを管理するために用いられる情報を記憶する領域を含んでいる。また、この発明では、処理系は、エージェントを実行した処理結果をログデータとして採取する。ログデータもオブジェクト（ログオブジェクト）として扱われる。採取されたログデータは、ディスクが利用可能であって、かつ、ログデータの記憶が要求されれば、ディスクに記憶される。また、ログデータの記憶が要求されていて、ディスクが利用可能でなければ、エージェントの管理者へ送信される。エージェントの管理者とは、エージェントを生成し、発行したユーザである。エージェントの管理者への送信は、エージェントを生成し、発行したユーザ端末へ送信することで行われる。一例として、ユーザ端末に送信する方法として、ユーザ端末のポート番号を宛先として送信することができる。また、他の例として、エージェントを発行したユーザのユーザIDを宛先として、ログを送信しても構わない。実際の送受信のやり方は、実現する送受信手段に依存する。この発明では、処理系が受信したエージェントは、処理系に備えられたエージェントから利用可能な不揮発記憶領域に記憶される。このエージェントの記憶操作は、処理系に予め備えられたロジックによりエージェントの実行前に行う。或いは、エージェント自身の手続き中に、処理系に到着したら、自分を登録するように記述しておき、処理系がエージェントの手続きを実行する中で、手続き中の記述に従って、エージェントを記憶させる。ユーザは、実行したいサービスに応じて、ユーザ端末101上でエージェントを生成する。ここで、ユーザは、エージェントに、あるデータベースの更新を2件実行させたいものとする。そこで、エージェントの実行を依頼する候補のサーバとして、図1に示すA, B, C, Dの4台を選択し、この4台の内のいずれかで実行するように手続きに記述して送出する。この実施の形態では、送出元のユーザ端末は、ユーザ端末のポートのアドレス（ポートアドレス）で識別される。

【0029】また、この実施の形態では、処理系は、エージェントを実行した実行状態を保持するオペレーションログを採取可能である。実際に、オペレーションログを採取するかどうかは、処理系の属するサーバの備えている機能や処理能力にも依存する。また、エージェントの中に、オペレーションログをとるかどうかを示す情報（ログ採取フラグ）を備え、このログ採取フラグの値により処理系に判断させてもよい。エージェントを送出する送出元（ユーザ、エージェント管理者）のポートアドレスを、エージェントの手続き中に登録しなければ、エージェントからのオペレーションログの報告はない。エージェントからのオペレーションログの報告とは、処理系で実行されたエージェントの実行状態を保持するオペレーションログやその他のステート情報をユーザ端末に返送することである。送出元のポートアドレスをエージェントの手続き中に登録しておけば、エージェントを生

成したユーザ、即ち、エージェント管理者は、オペレーションログを受信し、参照することができる。ユーザのエージェント送出機構、つまり、エージェント管理者がオペレーションログを受け取る実行例については後述する。

【0030】さて、ここでは、エージェント管理者への報告用のポートアドレスを登録していないので、エージェントは、エージェント管理者へのオペレーションの報告は行わず、オペレーションログはエージェントが実行される処理系の不揮発記憶領域に書き込みが行われる。

【0031】処理系によって、その属性に違いがある。エージェントは、処理系の属性の違いを処理系に問い合わせることにより知ることができる。エージェントは、処理系の属性を知り、また、処理系の属性データをエージェント内に蓄積することにより、エージェントが行うべき処理に合わせ、処理系を選択することができる。図4に、手続き中に、実行を依頼するサーバ名のリスト（ITINERARY、巡回サーバ名リスト、単に、リストともいう）を記述したエージェントの図を示す。ここでは、エージェントは、巡回サーバ名リストに記載されている順に移動するものとする。エージェントは、送出されると、巡回サーバ名リストの先頭にあるサーバAへ行く。正確には、このリストは図示しないエージェント送出機構が観察し、リストの先頭に登録されているアドレスに、該エージェントをサーバAにある処理系の通信機能を利用して送出する。サーバAにエージェントが到着し、処理系によってエージェントの処理を開始するための処理開始メソッド“start”が実行されると、エージェントは、属性情報を参照することによりサーバAの属性を知る。エージェントは、参照したサーバAの属性をエージェント内に格納し、エージェントの実行にふさわしいかを検討する。図5、301に示すのがサーバAの属性を示す属性情報（以降、サーバA属性情報ともいう）の一例である。図6に、サーバB属性情報、図7に、サーバC属性情報、図8に、サーバD属性情報の一例を示す。属性情報には、処理系の情報やサーバである計算機自身の属性情報等がある。

【0032】このエージェントが実行すべきメソッドは、データベース更新処理を含んでいるので、エージェントは、DBアクセスのできないサーバAでは実行できないと判断する。エージェントは、メソッドを実行できるサーバを求めて、リストの次のエントリであるサーバBに移動する。

【0033】エージェントは、サーバBに移動する。サーバAの場合と同様に、サーバの属性が実行にふさわしいかどうかを検討する。今度は、図6に示す302のサーバBでの処理系の属性情報にDBアクセスがあることから、処理可能と判断する。また、サーバB属性情報の“不揮発記憶部あり”から実行に先立ち、エージェント自身をサーバBの不揮発記憶領域に登録する。不揮発記

憶領域への登録は、サーバの属性情報を参照して、“不揮発記憶部あり”の場合に行う。また、エージェントを管理する管理情報に、エージェントを登録するか否かを示す不揮発記憶部格納要フラグを持たせておき、不揮発記憶部格納要フラグが“1”、即ち、“登録する”で、かつ、属性情報が“不揮発記憶部あり”的場合のみ格納させようにもよい。また、不揮発記憶部格納要フラグを更に分割し、エージェントを格納するか、オペレーションログを格納するかをアプリケーションの内容により個別に指定させてもよい。登録時には、図9に示す管理情報が、処理系により当該記憶領域の先頭部分に、図10に示すように、登録される。410は製品の識別番号であるマジック番号、412はエージェント機構のバージョン、401は不揮発記憶部格納要フラグ、402はオペレーションログのIDであるログオブジェクトIDである。また、403はエージェントを再実行する場合の再試行メソッドのアドレスとなる再試行メソッドオフセットである。また、414はエージェントに関連して記憶されるデータの件数を示すデータプール数である。図10に示す管理情報の構成は、501に該エージェントシステムでエージェントのIDである一意のオブジェクトID、502にオブジェクトタイプ、ステート503には、オブジェクトが有効か無効か、或いは、保存状態であるのかを示すステート情報を示される。ここでは、有効が示される場合を表している。504はオブジェクトの格納されるディスクオフセット位置(ディスクアドレス)である。ここでは、エージェントが登録されるので、該オブジェクトは、実行オブジェクトとして登録される。他のオブジェクトタイプとしては、ログ(ログオブジェクト)、データ等がある。510はオペレーションログを登録するためのエリアである。

【0034】自身の登録を済ませると、エージェントは処理を開始するが、オペレーションを開始する前に、図11、601にあるように、サーバB、“kai1ua@abc.opqrs.co.jp”という処理系に到着して、処理を開始した旨をオペレーションログとして登録する。その後、602から607まで各オペレーションの前にそれぞれのログを採取し、そのオペレーションを実行する。ここでは、データベーストランザクションが2回実行されており、それぞれ、例えば、update、delete、insert等のオペレーション、コミット開始、コミット終了で構成されている。オペレーションが終了すると、送出者であるユーザ端末にエージェントは戻る。

【0035】次に、サーバBに障害が発生した場合について説明する。同じ操作をユーザは行った。しかし、今回は、図12、701でサーバ到着を登録し、702登録後にサーバ更新操作を行なった後、703でコミット処理を行う旨を登録して、サーバBに障害が発生したとする。

【0036】サーバBのシステムアドミニストレータ(システム管理者)がサーバBをリブートすると、処理系も復旧する。処理系は、自身の復旧を終えると、不揮発記憶領域内に登録されているエージェントの再実行を図13の論理により行う。801でステートが有効であるオブジェクト全てをサーチする。有効なオブジェクトを発見したならば、802で処理系に登録されているオブジェクトを再びロードし、803で、図9に示したエージェントの管理テーブルをチェックする。ここで、図9の再試行メソッドオフセット403からオブジェクト内の再試行メソッド“restart”へのオフセットアドレスを知り、そのアドレスにアクセスして再試行メソッド“restart”を実行する。

【0037】エージェントが再実行を始めると、402にあるログオブジェクトIDがログオブジェクトへのポインタとなり、このポインタからエージェント自身がシステム障害前に行ったオペレーションのログ(図12)を得る。これから第1のデータベース更新のコミットオペレーションの完了が不定であることを知り、第1の更新のコミットから再試行する。再試行が正常終了すると、最終的に図11に示すログの実行結果と同等の結果を得て、エージェントは、ユーザ端末へ帰還する。このように、不揮発記憶領域にオペレーションログとエージェントオブジェクトを格納しておくので、オペレーションログを参照してエージェントの再実行が可能である。また、不揮発記憶領域にオペレーションログのみを格納しておく場合には、オペレーションログを元にユーザにエージェントの再発行を要求することで、エージェントの再実行を行うことが可能である。ユーザは、再発行が要求されたことにより、エージェントが正常に終了しなかったことを知り、再発行する。

【0038】次に、エージェントのオペレーションログを格納しない場合について説明する。ユーザは、オペレーションログを受けるポート(UDPヘッダーにより指定されるポートであり、IPアドレスを持つホストの内部であて先と送信元を区別するための番号)を生成する。その生成したポートのアドレスをエージェントに設定することで、そのエージェントのオペレーションログを受信することができる。いま、ここでポートアドレスを設定し、401の不揮発記憶部格納要フラグを“0”にセットし、エージェントの不揮発記憶部への格納を不要とし、前述した場合と同様に、このエージェントを送出する。エージェントは、前述した場合と同じ経路をたどり、サーバBの処理系で再び実行を開始し、前述の例でオペレーションログを格納するタイミングで、格納した場合と同じ内容のメッセージ(オペレーションログ)を管理者であるユーザ端末に送出する。

【0039】ここで、また、図12に示すような一連のメッセージを、メッセージ毎に管理者に送出したところでシステムに障害が発生したとする。予め設定してあ

る、一定の時間が経過しても、第1データベース更新のコミットメッセージが帰ってこないことから、正常に処理されていないと判断し、ユーザは、エージェント削除を削除メッセージを送出することによって行う。念のため説明すると、ユーザは、エージェントからのメッセージを受けとることによって、エージェントの現在のポートアドレスを知ることができる。そのエージェントの現在のポートアドレスに対して削除メッセージを送出することで、エージェント削除をエージェントが処理されていた処理系で実行できる。

【0040】ユーザは、受け取ったオペレーションログから引き続いて実行すべき処理のリストを再作成し、その処理のリストを記述したエージェントを新たに生成する。生成したエージェントは、先のエージェントの処理を引き続いて行うので、先のエージェントと同じ巡回先リストに沿って移動する。オペレーションログ受信ポートアドレスは、サーバBだった。この時点で、まだサーバBは障害から復旧していないので、次にリストに登録されているサーバCが移動先となる。サーバCに到着すると、エージェントは、図7に示すサーバCの属性情報をチェックする。サーバCは、“DBアクセスあり”なので、エージェントはサーバC上の処理系でその実行を開始し、後続処理を行なって帰還する。このように、オペレーションログをユーザ端末に送信することによっても、エージェントの再実行が可能である。この場合、エージェントの再実行は、サーバBではなく、サーバCで行われるので、サーバBで処理系の再実行を行うと、二重に処理することになってしまう。だが、ここでは、サーバBが再びポートアップしたとき、先に送られたエージェントステートは、削除されているので一切残っていない。このように、削除することにより二重処理の発生を防ぐことができる。また、オペレーションログは、ユーザ端末に送出されており、不揮発記憶部には格納されていないので、問題はない。

【0041】前述した実施の形態では、エージェントとログの両方を格納する場合、及びエージェントを格納せず、ログのみを格納する場合について説明したが、エージェントのみを格納しても構わない。格納されているデータがエージェントであれ、オペレーションログであれ、格納されているデータに対する削除メッセージを発行すれば、同様の効果となる。

【0042】次に、オペレーションログを残す場合について説明する。さて、ユーザがエージェントを利用して行ったオペレーション、或いは、実行中のオペレーションを取り消したい場合、ステート情報を受信するポートアドレスをエージェント内に登録しないとき、即ち、オペレーションログをユーザ端末に送出せず処理系にそのオペレーションログを残す場合は、以下のように取り消しを行う。

【0043】取り消したいエージェントと同じ巡回サー

バリストを持った取り消し用エージェントを発行することにより、取り消しを行う。取り消し用エージェントが処理系に入ったならば、取り消したいエージェントのオブジェクトIDで不揮発記憶部内をサーチし、取り消したいエージェントが保存したそのエージェント自身に関する情報を取り消し用エージェント内に保存する。保存するのは、後述する処理でその情報を使用するためである。サーチした情報のステート503に“有効”が設定されていれば、先のエージェントは実行中である、処理系が予め用意する所定のシグナルにより、実行中である先のエージェントを停止させ、停止後、先のエージェントを削除する。先のエージェントを削除した上で、不揮発記憶領域内の先のエージェントが登録したエージェント自身、オペレーションログのステート503のフィールドに“無効”をセットし、取り消し用エージェント内に先ほど保存した先のエージェントのオペレーションログを元に、オペレーションの訂正処理、即ち、データベースのロールバックを行う。

【0044】不揮発記憶領域内に保存されているステート503が“保存”であったなら、既に先のエージェントはその処理系での処理を終えて、次の目的地へ移動している。その場合には、保存されている先のエージェント自身及びそのオペレーションログを削除して、データベースのロールバック処理を行う。

【0045】以上のように、サーバ巡回リストとオペレーションログを用いてエージェントを追跡し、取り消し処理を終えた取り消し用エージェントは、ユーザの元に帰還する。

【0046】取り消したいエージェントがステート情報を受信するポートアドレスをエージェント内に登録してあった場合、即ち、エージェントがオペレーションログをユーザ端末に送出し、処理系にはそのオペレーションログを残さない場合は、ユーザがそのステート情報を（オペレーションログ）を判断し、処理系に削除メッセージを送り、ロールバック処理エージェントを再発行することによってエージェントの取り消しを行う。

【0047】次に、エージェントが自分を実行してもらう処理系を選択する手順について説明する。エージェントは、自分を実行するために、必要な実行条件を手続き中に予め持っているものとする。まず、1つの方法について説明する。エージェントが処理系に到着すると、その処理系の属性情報を参照し、自分の実行条件と照合する。照合した結果、処理系の属性情報と自分の実行条件が一致していれば、その処理系で実行を開始してもらう。一致していないければ、エージェントは次の処理系へ移動する。エージェントは、到着した処理系の属性情報を同様にチェックし、実行可能な処理系にたどり着くまで、移動を続ける。エージェントの移動については、前述したように、エージェントの管理者が、予め巡回先リストをエージェントの手続き中に記述しておき、巡回先

リストに従って、移動するやり方がある。移動する順番は、巡回先リストに記載した順番とする。又は順不同でもよい。或いは、手続き中で細かく指定してもよい。或いは、手続き中に選択肢を用意して、移動実行時のエージェントと処理系に判断させててもよい。また、手続き中で細かく指定せずに、到着した処理系で実行可能でなかったとき、代替の処理系を教えてもらってよい。また、ある処理系でエージェントの実行が完了したとき、その処理系から次に移動する処理系の候補を教えてもらい、その候補を移動先としてもよい。或いは、これらの移動方法の組み合わせでもよい。

【0048】次に、複数の処理系の属性情報を収集して、最適な処理系を選択する場合について説明する。エージェントが属性情報を収集する処理系の数を、仮に5とする。これは、一例であり、10でも20でも任意の数とする。エージェントは、ネットワークを移動して、到着した処理系の属性情報を得る。エージェントは、指定された処理系の数になるまで、属性情報の収集を繰り返す。属性情報を収集した処理系の数が指定数（ここでは、5）に達すると、エージェントは収集した各処理系の属性情報と自分の実行条件に合うものを選ぶ。1つの条件の一一致した処理系が複数あるときは、2つ目の条件で更に選ぶ。これを繰り返して、実行条件が一致する処理系が複数あるときは、例えば、処理能力の高いものを選択するというように、優先順位を設定しておいてもよい。また、処理能力の高低でなく、移動のしやすさで処理系を選択してもよい。その際、移動のしやすさが属性情報から判断できるようになっていることはもちろんである。また、リストの記載順で処理系を選択してもよい。また、デフォルトの優先順位を予め設定しておき、実行条件で絞り込めない場合には、その優先順位を用いてもよい。

【0049】次に、図14、図15を用いて属性情報サーバについて説明する。図14は、エージェントを処理可能な複数の処理系の属性情報を、エージェントに提供する属性情報サーバを示す図である。図において、属性情報サーバ1106は、不揮発記憶部に属性情報1116を記憶しておき、その属性情報1116をエージェントに参照させる。属性情報1116の一例を、図15に示す。属性情報1116の内容は、前述した図5～図8に示したサーバA～サーバDの属性情報と同一である。だが、それらの属性情報を1つのサーバ上に集めて属性情報サーバとし、属性情報サーバ（この例では、サーバA）が要求に応じて、他のサーバの属性情報を一度に提供できることが属性情報サーバの特徴である。属性情報サーバのサービスを受けることで、エージェントは、実際に最適なサーバを短時間で、ネットワークに負荷をかけずに選択できる。

【0050】

【発明の効果】本発明によるエージェント方式を用いる

ことにより、エージェントに依頼する処理を確実に、実行又は削除することができるので、従来、処理が不確実であったエージェント方式によっても、ミッションクリティカルな処理を実行することができる。

【0051】この発明によれば、エージェントの実行状態を示すオペレーションログを残すので、実行状態を知ることができる。

【0052】また、この発明によれば、処理系に、受信したエージェントの記録を残すことができる。

【0053】また、この発明によれば、ユーザ端末側で、発行したエージェントのオペレーションログを受け取るので、実行状態を知ることができる。

【0054】また、この発明によれば、正常終了しなかったエージェントの再実行を、処理系側だけで行なえる。

【0055】また、この発明によれば、処理系側で、正常終了しなかったエージェントの再実行を起動できる。

【0056】また、この発明によれば、ユーザ端末主導で、エージェントを再発行することにより、エージェントの再実行を行える。

【0057】また、この発明によれば、エージェントを削除し、二重処理を防ぐことができる。

【0058】また、この発明によれば、エージェントの追跡を行うので、発行したエージェントを管理できる。

【0059】また、この発明によれば、不要なログを残すことが無くなる。

【0060】この発明によれば、エージェントをより適した処理系で実行するようエージェント自身が選択できる。

【0061】この発明によれば、ユーザ端末は、自身が発行したエージェントの実行状態を知ることができる。

#### 【図面の簡単な説明】

【図1】 本発明を適用したシステム構成例を示す図。

【図2】 本発明の不揮発記憶部の構成例を示す図。

【図3】 本発明で用いるエージェントの構成例を示す図。

【図4】 本発明で用いるエージェントの構成例を示す図。

【図5】 本発明の処理系属性の例を示す図。

【図6】 本発明の処理系属性の例を示す図。

【図7】 本発明の処理系属性の例を示す図。

【図8】 本発明の処理系属性の例を示す図。

【図9】 エージェントオブジェクト管理情報例を示す図。

【図10】 本発明の不揮発記憶部への管理情報の記憶例を示す図。

【図11】 本発明のオペレーションログ例を示す図。

【図12】 本発明のオペレーションログ例を示す図。

【図13】 本発明の処理系におけるエージェント再実行論理例を示す流れ図。

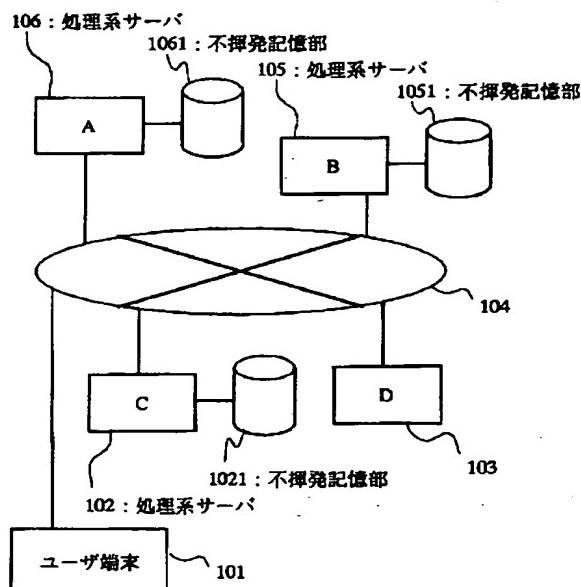
- 【図14】 本発明の属性情報サーバを示す図。
- 【図15】 本発明の属性情報サーバの属性情報の一例を示す図。
- 【図16】 従来技術によるコンピュータシステムの構造を示す図。
- 【図17】 従来技術によるリモートプログラミングを使用する方法の流れ図。
- 【図18】 従来技術によるリモートプログラミングを

具体化するネットワークを示す図。

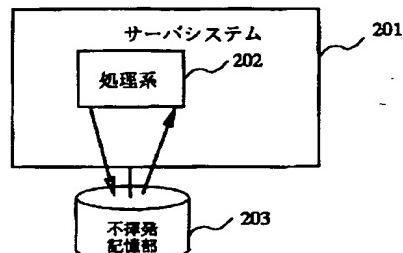
【符号の説明】

202 処理系、203 不揮発記憶部、301～303 処理系属性例、401 不揮発記憶部格納要フラグ、402 ログオブジェクトID、403 再試行メソッドオフセット、501 オブジェクトID、502 オブジェクトタイプ、503 ステート、504 ディスクオフセット位置。

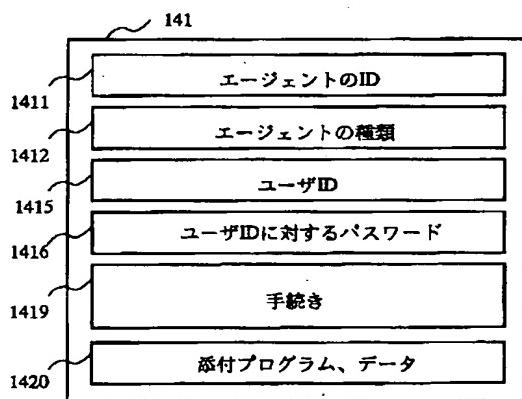
【図1】



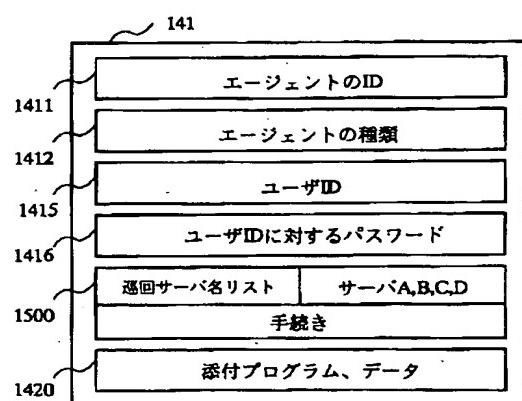
【図2】



【図3】



【図4】



【図5】

不揮発記憶部あり
DBアクセスなし
処理能力大
⋮
サーバA属性情報

301

【図6】

不揮発記憶部あり
DBアクセスあり
処理能力小
⋮
サーバB属性情報

302

【図7】

不揮発記憶部あり
DBアクセスあり
処理能力大
⋮
サーバC属性情報

301

【図8】

不揮発記憶部なし
DBアクセスあり
処理能力大
⋮
サーバD属性情報

304

【図9】

マジック番号
バージョン
不揮発記憶部格納要
ログオブジェクトID
データプール数
⋮
再試行メソッドオフセット

410

412

401

402

414

403

403

【図10】

501	502	503 : ステート	504
オブジェクトID	オブジェクトタイプ	有効	オフセット
510			

501

502

503 : ステート

504

510

【図11】

```

601 got into "kaihua@abc.opqrs.co.jp" 97:1:30:18:55:10
602 Tr1 started 97:1:30:18:55:11
603 Tr1 commit started 97:1:30:18:55:21
604 Tr1 committed 97:1:30:18:55:33
605 Tr2 started 97:1:30:18:55:33
606 Tr2 commit started 97:1:30:18:55:44
607 Tr2 committed 97:1:30:18:55:51
608 got out from "kaihua@abc.opqrs.co.jp" 97:1:30:18:55:51

```

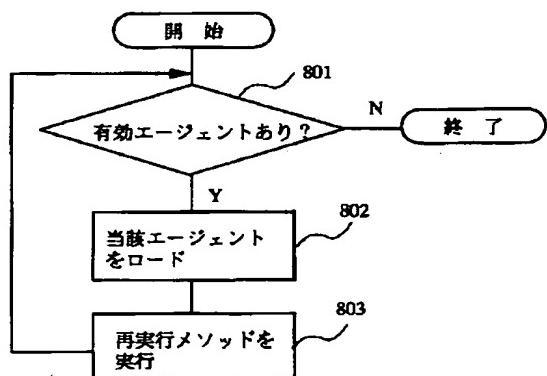
【図12】

```

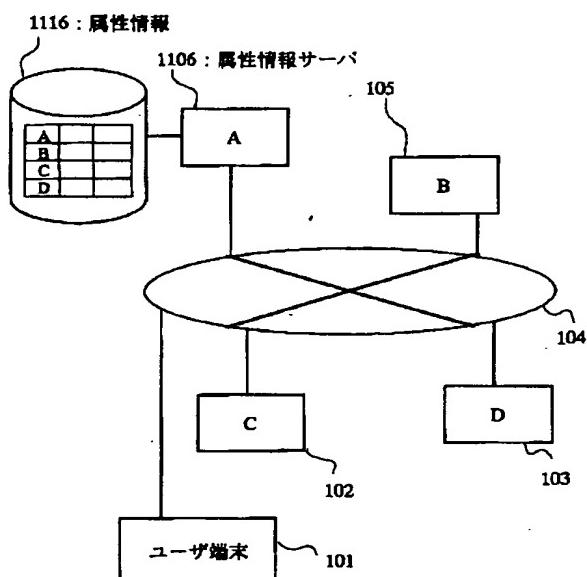
701 got into "kaihua@abc.opqrs.co.jp" 97:1:30:18:55:10
702 Tr1 started 97:1:30:18:55:11
703 Tr1 commit started 97:1:30:18:55:21

```

【図13】



【図14】

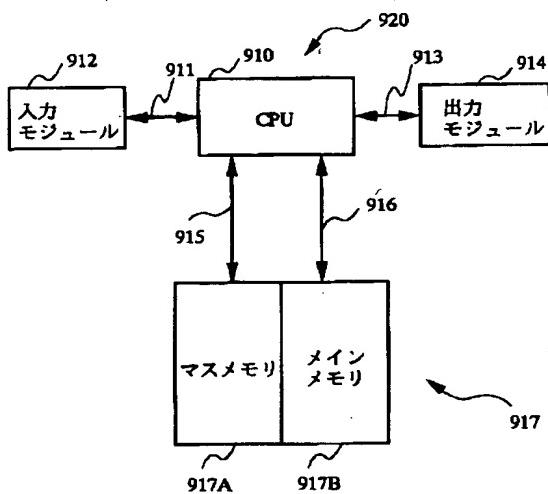


【図15】

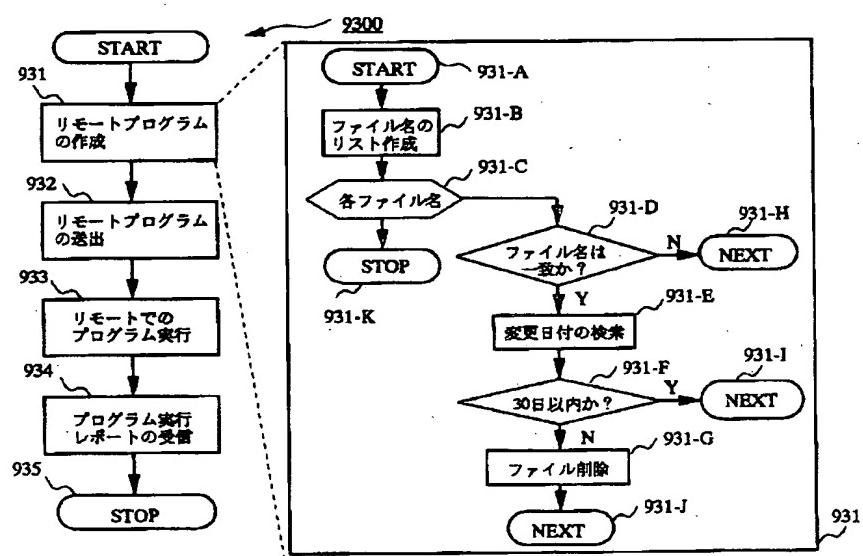
1116

属性情報				
サーバ識別子	不揮発記憶部	DBアクセス	処理能力	-----
A	あり	なし	大	-----
B	あり	あり	小	-----
C	あり	あり	大	-----
D	なし	あり	大	-----

【図16】



【図17】



【図18】

